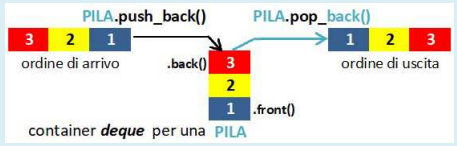
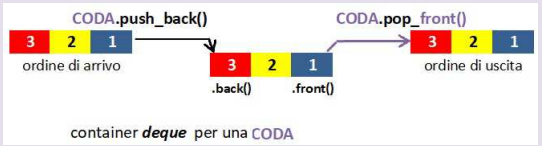
Esempio di gestione di una PILA e di una CODA mediante *container deque* (double-ended queue)

prof.ssa P.Grandillo



pila_deque.cpp (esempio: pila di pratiche da elaborare)



coda_deque.cpp (esempio: fila di persone da servire)

```
#include <iostream>
#include <fstream>
  using namespace std;
#include <deque>

  deque <int> PILA;

  ifstream finp ("finp_P_C.txt");

void scriviPila (){
    cout << "\n\nin PILA pratiche: ";
    cout << PILA.size();
    cout << " - prima: "  << PILA.front();
    cout << " - ultima: " << PILA.back() <<endl;

  deque<int>::iterator i;
    for( i = PILA.begin(); i != PILA.end(); i++ )
       cout<< *i <<' ';
}
int Push  (int pra) {

    PILA.push_back(pra);
    return 0;
}
int Pop  (int& praT) {
    if ( PILA.empty() ) {
      cout <<"\n\n-----------------------------------\n";
      cout <<   "errore Pila vuota, nessuno da servire";
      return -1;
    }
    scriviPila ();
    praT = PILA.back();
    PILA.pop_back();
    return 0;
}
```

```
#include <iostream>
#include <fstream>
  using namespace std;
#include <deque>

  deque <int> CODA;

  ifstream finp ("finp_P_C.txt");

void scriviCoda () {
    cout << "\n\nin CODA persone : ";
    cout << CODA.size();
    cout << " - prima: "  << CODA.front();
    cout << " - ultima: " << CODA.back() <<endl;

  deque<int>::iterator i;
    for( i = CODA.begin(); i != CODA.end(); i++ )
       cout<< *i <<' ';
}
int Push  (int per) {

    CODA.push_back(per);
    return 0;
}
int Pop  (int& perT) {
    if ( CODA.empty() ) {
      cout <<"\n\n-----------------------------------\n";
      cout <<   "errore Coda vuota, nessuno da servire";
      return -1;
    }
    scriviCoda ();
    perT = CODA.front();
    CODA.pop_front();
    return 0;
}
```

Esempio di gestione di una PILA e di una CODA mediante *container deque* (double-ended queue)

prof.ssa P.Grandillo

```
int main(){

  int PRA, praPila;          //-- PRA pratica letta da file
  int N, esito=0;           //-- praPila prelevata da PILA

    finp >> N;

    for (int i=0; i<N && esito==0; i++) {

        finp >> PRA;

        if (PRA > 0)           //--- arriva  pratica
            esito = Push(PRA);
        else                   //---elaborata pratica
        {   esito = Pop (praPila);
            if (esito == 0)
                cout << "\nconclusa pratica: "<< praPila;
        }
    }

    if (esito == 0)

      if ( PILA.size() > 0 ) {     //-- condizioni equivalenti

        cout << "\n\n-------------------------\n";
        cout <<     "errore: Pila ancora piena";
        scriviPila ();
      }
      else
        cout <<"\n\n OK - non ci sono altre pratiche nella Pila";
    cout << endl;
    finp.close();
    return 0;
}
```

```
int main(){

  int PER, perCoda;          //-- PER persona letta da file
  int N, esito=0;           //-- perCoda prelevata da CODA

    finp >> N;

    for (int i=0; i<N && esito==0; i++) {

        finp >> PER;

        if (PER > 0)           //---arriva persona
            esito = Push(PER);
        else                   //--- esce persona
        {   esito = Pop (perCoda);
            if (esito == 0)
                cout << "\nservita persona : "<< perCoda;
        }
    }

    if (esito == 0)

      if ( ! CODA.empty() ) {     //-- condizioni equivalenti

        cout << "\n\n-------------------------\n";
        cout <<     "errore: CODA ancora piena";
        scriviCoda ();
      }
      else
        cout <<"\n\n OK - non ci sono altre persone nella Coda";
    cout << endl;
    finp.close();
    return 0;
}
```

NOTA: anche per il container deque (coda a due ingressi) le condizioni **nome.size() > 0** *e* **! nome.empty()** *sono equivalenti*

per approfondimenti consultare :

https://www.cplusplus.com/reference/deque/deque/

Esempio di gestione di una PILA e di una CODA mediante *container deque*

casi di Test

prof.ssa P.Grandillo

Entrambi i programmi leggono dal File di Testo il numero N di ingressi ed uscite dalla **PILA**/**CODA**; poi leggono gli N valori successivi: un numero positivo indica il progressivo della **pratica**/**persona** in arrivo (1 2 ...) che va inserita nel ***container deque*** mediante il metodo ***.push_back()***, il valore **-1** indica che una **pratica**/**persona** è stata **elaborata**/**servita** e che, dalla **PILA** può essere eliminato **l'ultimo elemento dalla fine** mediante il metodo ***.pop_back()*** mentre la **CODA** può scorrere **eliminando il primo elemento dall'inizio** mediante il metodo ***.pop_front()***.

OUTPUT con *pila_deque.exe*	**OUTPUT con *coda_deque.exe***
caso 1 – file di input corretto →	**8** 1 2 -1 3 4 -1 -1 -1
Prompt dei comandi >>pila_deque.exe in PILA pratiche: 2 - prima: 1 - ultima: 2 1 2 conclusa pratica: 2 in PILA pratiche: 3 - prima: 1 - ultima: 4 1 3 4 conclusa pratica: 4 in PILA pratiche: 2 - prima: 1 - ultima: 3 1 3 conclusa pratica: 3 in PILA pratiche: 1 - prima: 1 - ultima: 1 1 conclusa pratica: 1 OK - non ci sono altre pratiche nella Pila >>	Prompt dei comandi >>coda_deque.exe in CODA persone : 2 - prima: 1 - ultima: 2 1 2 servita persona : 1 in CODA persone : 3 - prima: 2 - ultima: 4 2 3 4 servita persona : 2 in CODA persone : 2 - prima: 3 - ultima: 4 3 4 servita persona : 3 in CODA persone : 1 - prima: 4 - ultima: 4 4 servita persona : 4 OK - non ci sono altre persone nella Coda >>
caso 2 – input errato (uscite inferiori a ingressi) →	**6** 1 2 3 4 -1 -1
Prompt dei comandi >>pila_deque.exe in PILA pratiche: 4 - prima: 1 - ultima: 4 1 2 3 4 conclusa pratica: 4 in PILA pratiche: 3 - prima: 1 - ultima: 3 1 2 3 conclusa pratica: 3 -------------------------- errore: Pila ancora piena in PILA pratiche: 2 - prima: 1 - ultima: 2 1 2 >>	Prompt dei comandi >>coda_deque.exe in CODA persone : 4 - prima: 1 - ultima: 4 1 2 3 4 servita persona : 1 in CODA persone : 3 - prima: 2 - ultima: 4 2 3 4 servita persona : 2 -------------------------- errore: CODA ancora piena in CODA persone : 2 - prima: 3 - ultima: 4 3 4 >>
caso 3 – input errato (uscita precede ingresso) →	**6** 1 2 -1 -1 -1 3
Prompt dei comandi >>pila_deque.exe in PILA pratiche: 2 - prima: 1 - ultima: 2 1 2 conclusa pratica: 2 in PILA pratiche: 1 - prima: 1 - ultima: 1 1 conclusa pratica: 1 -------------------------------------- errore Pila vuota, nessuno da servire >>	Prompt dei comandi >>coda_deque.exe in CODA persone : 2 - prima: 1 - ultima: 2 1 2 servita persona : 1 in CODA persone : 1 - prima: 2 - ultima: 2 2 servita persona : 2 -------------------------------------- errore Coda vuota, nessuno da servire >>